

AD-A126 171

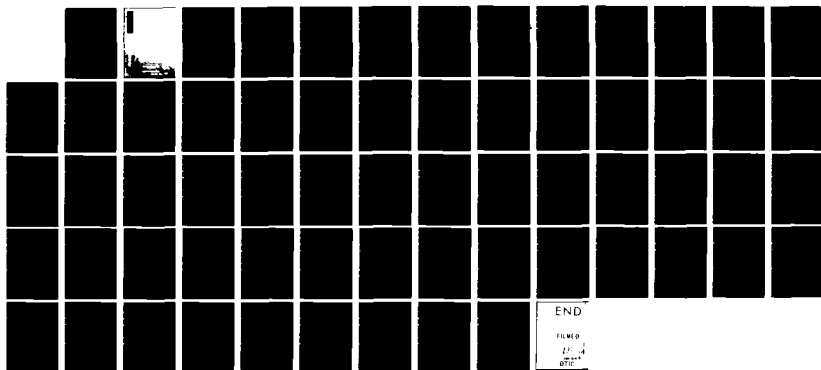
RESTRAINED LEAST-SQUARES PROGRAM (RESLSQ) USERS MANUAL
(U) NAVAL RESEARCH LAB WASHINGTON DC
J L FLIPPEN-ANDERSON ET AL. 28 MAR 83 NRL-MR-5842

1/1

UNCLASSIFIED

F/G 9/2

NL

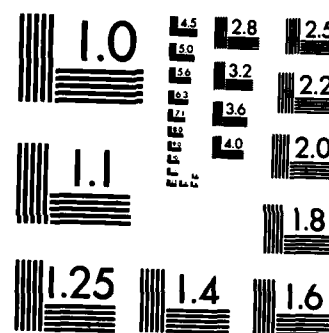


END

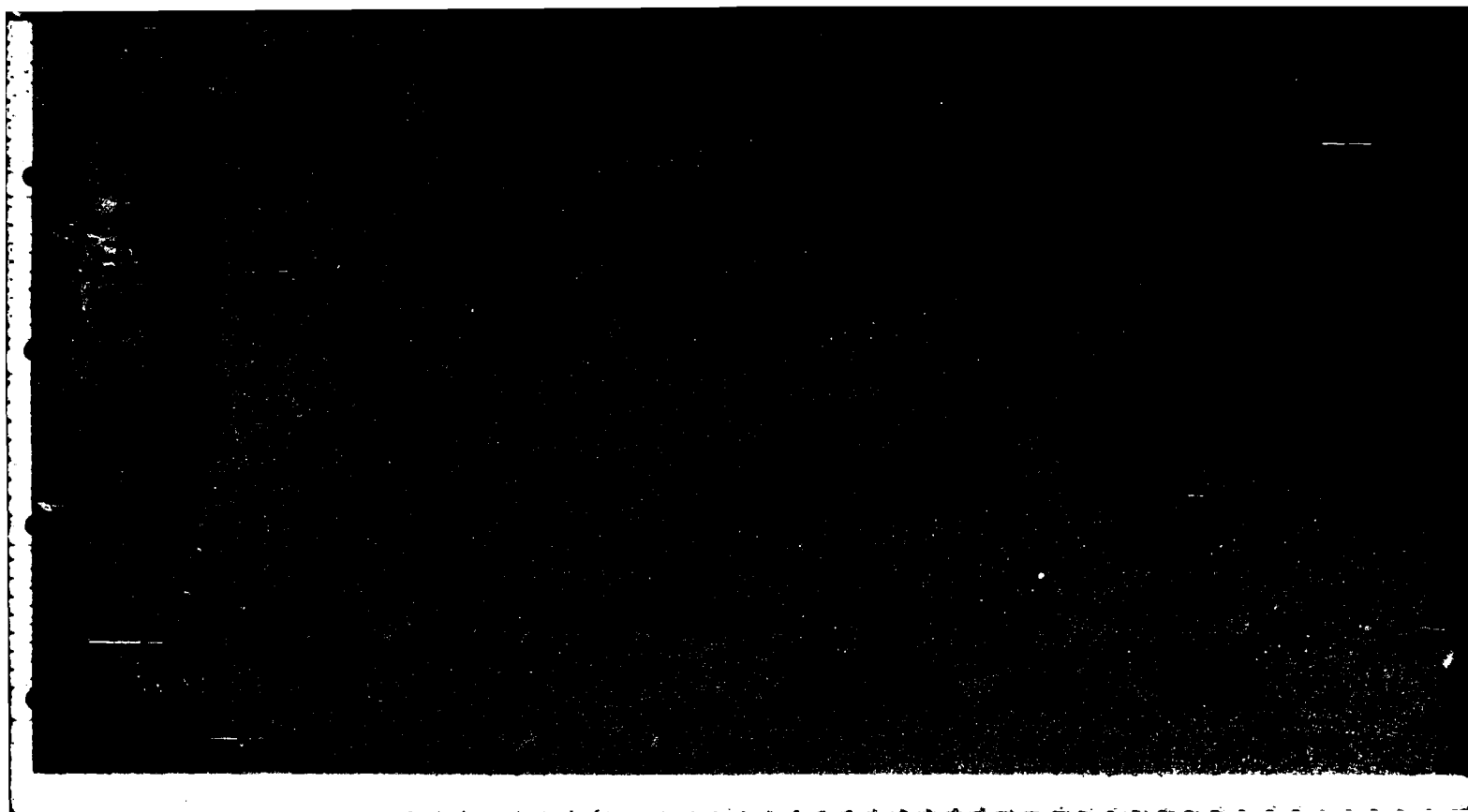
FILMED

17-4

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 5042	2. GOVT ACCESSION NO. AD-4126171	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RESTRAINED LEAST-SQUARES PROGRAM (RESLSQ) USERS MANUAL		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. L. Flippen-Anderson, R. Gilardi and J. H. Konnert		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D. C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61153N; RR011-03-41; 60-0303-0-3
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE March 28, 1983
		13. NUMBER OF PAGES 62
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) (RESLSQ)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A restrained least-squares program package is available for analysis of x-ray diffraction data from small and medium sized molecules. The package consists of three basic programs. The first, SCATT, prepares a binary data file which contains scattering factor information for each reflection. The second, MODEL, constructs the "ideal" molecule towards which the experimental model is refined. The third, RESLSQ, performs the least-squares refinement using conjugant gradient sparse-matrix methods.		

DD FORM 1473
1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

Introduction	1
Program SCATT	3
Program MODEL	5
Appendices	17
Glossary	26
Program RESLSQ	29
Introduction	30
Control cards (input)	34
Subroutine descriptions	40
File information	50
Machine specific code	53
Subroutine arrays	55
References	60



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

RESTRAINED LEAST-SQUARES PROGRAM (RESLSQ) USERS MANUAL

INTRODUCTION:

This refinement package consists of three basic programs. The first program, "SCATT", prepares a binary reflection file which contains scattering factor information for each reflection. We have found that this saves time in the structure factor calculation by eliminating table look-up and interpolation.

The second, "MODEL", constructs an "ideal" molecule from a connectivity matrix, covalent radii, and bond angles supplied by the user and then writes files of "ideal" interatomic distances towards which the "real" model will be restrained during refinement.

The third program, "RESLSQ", performs the least-squares refinement. It imposes elastic bounds on all of the key geometry-defining interatomic distances (Konnert, 1976; Hendrickson & Konnert, 1979) which were identified by MODEL (or specified by the user) and also restrains the thermal factor parameters in such a manner that the relative motion for pairs of stereochemically related atoms does not exceed physically reasonable values (Konnert & Hendrickson, 1980). Thermal parameters used in RESLSQ are of the same form as are the betas in ORFLS (Busing, et al, 1975). Groups of atoms may also be elastically restrained towards coplanarity. How tightly any particular class of restraints is enforced is at the user's discretion and may be changed during the course of a refinement.

The restrained least-squares procedure uses a greatly reduced ("sparse") matrix; shifts in parameters are derived from

Manuscript approved January 4, 1983.

this matrix not by inversion but by the iterative conjugate gradient method as described by Konnert (1976). For a 100 atom problem the "sparse" matrix contains approximately 7% of the number of elements needed for a "full" matrix refinement. With increasing numbers of atoms the size of the matrix increases linearly rather than exponentially as it does in full-matrix methods.

The method has already proven to be extremely useful in the refinement of macromolecules (Konnert, 1976; Sielecki, Hendrickson, Broughton, Delbaere, Brayer & James; 1979; Furey, Wang, Yoo & Sax, 1979; Honzatko & Lipscomb, 1982; Reese & Lipscomb, 1982). The RESLSQ program itself has been used on several small and mid-sized molecules with up to as many as 213 atoms in the asymmetric unit (Flippen-Anderson, 1982; Flippen-Anderson et al, 1982; 1983; Karle et al, 1983) and has become the standard refinement program used in our laboratory.

PROGRAM SCATT

(Prepares reflection input file for RESLSQ - file 20)

Written by

Wayne Hendrickson

Laboratory for the Structure of Matter

Naval Research Laboratory

Washington, D. C. 20375

Program SCATT prepares the binary reflection file read by RESLSQ (file 20). It contains scattering factor information for each reflection.

CONTROL CARDS:

Type 1
(one card)

Real Cell Dimensions (6F8.2)

ACELL
BCELL
CCELL
ALPHA
BETA
GAMMA

Type 2
(one card)

of Kinds of Atoms (I5)
NS

Type 3
(NS cards)

Scattering factor tables (4x,I4,9F8.5)

J = scattering factor order -
i.e. if J = 1 for carbon scattering
factors then FMULT (FILE 10 - atom
coordinate input to RESLSQ) must = 1.0
for carbon.

Cols 9-80 carry the 9 scattering factors
needed for atom type J.

Ref: Acta Cryst. A24, 321 (1968)
or
Int. Tables - Vol. IV
Table 2.2B - Page 99

SCATT also reads H,K,L, FOBS and SIG values from file 10
(not related to file 10 for RESLSQ)

PROGRAM MODEL

WRITTEN BY:

PAUL B. ANDERSON

FOR:

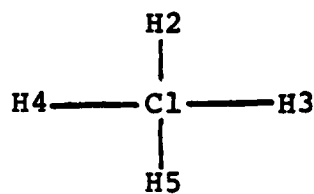
JUDITH L. FLIPPEN-ANDERSON
LABORATORY FOR THE STRUCTURE OF MATTER
NAVAL RESEARCH LABORATORY
WASHINGTON, D.C. 20375

1. Introduction. The MODEL program accepts a description of the bonds comprising a molecule and outputs idealized distances between various atom pairs.

2. Parsing the Input. The molecule description in the syntax shown by the examples of Figure 1 is processed by the PARSE subroutine. PARSE is repeatedly called by the BUILD subroutine until the input is completely read. PARSE uses a number of supporting routines to analyze the input characters. The characters are treated as a continuous stream without regard to card boundaries. Routine NXCHAR provides the characters, one-by-one, to the parsing subroutine. PRCHAR is used in some cases to return characters. PARSE controls the entire scan with a method based on a finite state model of the input grammar. The principal advantages of this are that it is largely table driven, guaranteeing uniformity and easy maintenance, and that a relatively complex structure can be easily handled. The array TRANS, shown in Appendix A, and the variable CURST constitute the core data structure of PARSE. CURST gives the current "state" of the parse (see Appendix B). PARSE analyzes the input and determines the lexical class (see Figure 2) of the next characters using the routines ATOMIN and LISTIN. The transition table TRANS is accessed with the current state and the lexical class of the input to determine what the next state will be and what action is to be taken. Some of the actions in PARSE result in data being returned to BUILD.

C1:S(0.77,109.5)-H2:S(0.23,0.0),-H3:S,-H4:S,-H5:S

represents



C1:S-H6:S,-H7:S,-H8:S

C2:I(0.695,120.0)-C1:S,-O3:S(0.66,109.5)

C2:D(0.665,120.0)-O4:D(0.55,120.0)

O3:S-H5:S

represents

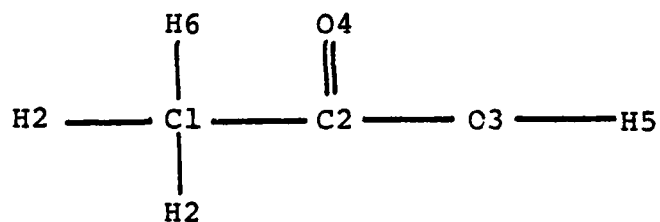


Figure 1. Input Syntax Examples

<u>Lexical Item</u>	<u>Class</u>
Complete atom eennn:b	1
Incomplete atom :b	2
List (r_1, r_2)	3
Comma or space	4
Dash	5
End of file	6
Other	7

Figure 2. Lexical Classes

3. Data Structures Created by BUILD Phase. Two data structures are created by subroutine BUILD. The first structure is the internal representation of the bond information supplied in the input. The structure accommodates atoms numbered from 1 through 200 and up to 400 bonds.

Access to the bond information is through the atom entries shown in Figure 3. The bond entries are organized as parallel arrays, storing all of the information on atom "I" in the Ith subscript position. Thus, the character name, bonding angle and bond list pointer are stored in array elements ATOML(I), AANGLE(I) and BOND(L(I), respectively. The atom number is the key to access of the atom list. The bond list may be accessed using the subscript supplied in the BOND(L entry for the atom of interest. The bond list is organized as (five) parallel arrays in the same manner as the atom list. Figure 4 details the contents of the bond list entries. The bond entries are chained together as a linked list. The link between bond entries is given in the BOND(NX part of a bond entry. The BOND(NX value is the subscript value of the next bond entry for the atom. The list is terminated by a zero link. To speed up processing of non-bonded distances, bonds are always stored twice. If atom "A" is bonded to atom "B", an entry is made in atom A's bond list to reflect this. A corresponding entry is made in atom B's bond list (showing a bond to A). To distinguish the bonds input from these generated, duplicate bonds, the BOND(NX values of the duplicates are negated.

ATOML	Two character representation of the element symbol, e.g. 'CL'.
AANGLE	Angle between bonds for this atom.
BONDL	Subscript of first bond entry for this atom (the origin atom).

Figure 3. Atom List Entries

ATOMNO	Number of atom being bonded to (target atom).
BNDTY1	1 character bond type for origin atom.
BNDTY2	1 character bond type for the target atom.
BDIST	Bond distance - set to 0 in BUILD, computed in BONDD.
BONDNX	Index of next bond entry for origin atom, negated if this bond was not input (i.e., a bond created by BUILD).

Figure 4. Bond List Entries

The second structure built during subroutine BUILD's execution stores the bond distances and angles. Ordinarily this structure is retrieved from the unformatted file on unit 1. User input may update or add to this data if desired. The final structure is printed and output to unit 2 before processing completes. To add to or update the structure, subroutine BUILD uses the auxiliary routine ADDRAD. The structure is similar to the atom and bond structures described above except that the element list (ELEM) entries are not ordered in any way. Thus, to find a particular bond type for an element, the element is first located in the element list. RADL, a parallel array, then gives the index of the first bond type entry. This is shown in Figure 5. The bond type entries are similar in organization to the bond list entries of the bond data structure. Bond type entries have the structures shown in Figure 6. (Note: Structure here is used in a programming sense and does not refer to the "model structure" being described by the restraint parameters.)

4. Bonded Distances. The bond distances are computed by routine BONDD once the input has been completely scanned. BONDD also establishes the bond angles used in subroutine NBONDD to compute the non-bonded distances. Bond distances are determined by the bonding radii of the two atoms involved. These bonding radii vary for each element depending on the bond type (e.g. single or double). The bond distance is computed simply as the sum of the two radii. The bond distances are output to unit 26 in the format shown in Figure 7. Each bond distance is stored in the bond entry (BDIST) for use by NBONDD.

ELEM
RADL

2 character symbol for element,
e.g. 'CL'.

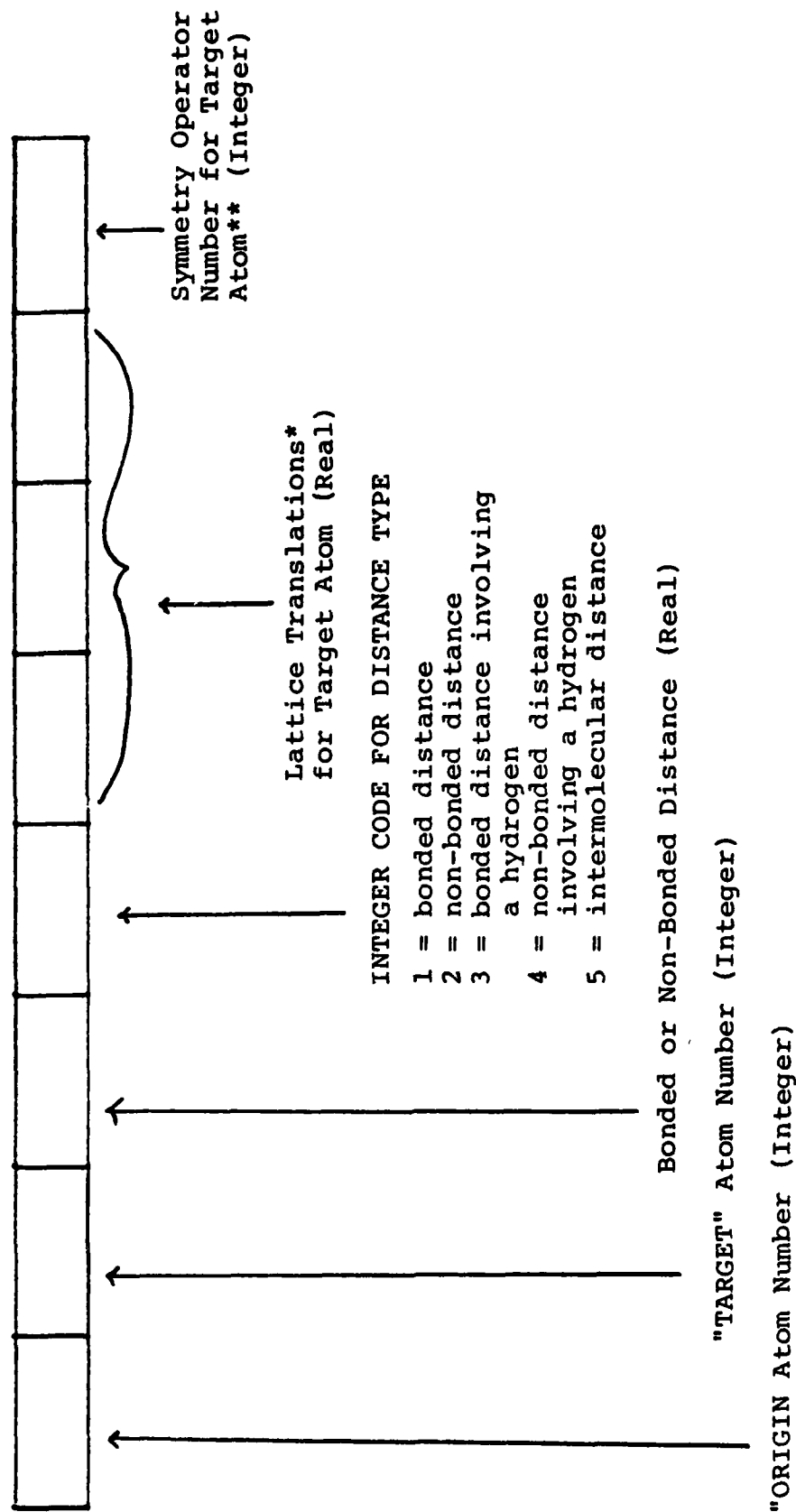
Subscript of first bond type entry
for this element.

Figure 5. Element List

BTYPE	1 character bond type.
BONDR	Radius for a bonded distance of this type.
ANGLE	Angle for bonds (used to set array AANGLE).
BRLNXT	Subscript of next bond type entry. 0 if no more entries.

Figure 6. Bond type Entries.

Each record is produced by unformatted I/O in the following format:



* All three lattice translations are set to 5.0
 ** Set to 1.

Figure 7. Unit 26 Output Format For the Identity Molecule

5. Non-Bonded Distances. Non-bonded distances are computed using the law of cosines for any two distinct atoms bonded to a common atom. The separation of these atom pairs is determined by the bond distances and the bond angles created by BONDD. Non-bonded distances are output to unit 26 in the format given in Figure 7.

6. The user may supply specific distance restraint information. Distance information would be input through unit 16 and added by SUBROUTINE PREFIX to unit 26 output.

7. Block Data Output. The bonding radii and angles used for BONDD and NBONDD are kept in common/RADII/. This common is created by the user input and/or read from unit 1 when the program first begins execution. Once the program has completed the output for unit 26 as described above, the RADII common is output using unformatted I/O to unit 2 so that it may be retained for subsequent use (as unit 1). The bond descriptions are thus easily expandable and can retain their values across program executions. It is intended that the tables in RADII reach a point where additions are few and those that do occur are for special situations.

APPENDIX D contains file descriptor information, details on control cards necessary to execute the program and a glossary of terms which are used in the COMMON blocks.

Appendix A

TRANS(*,*,1) - TRANSITIONS

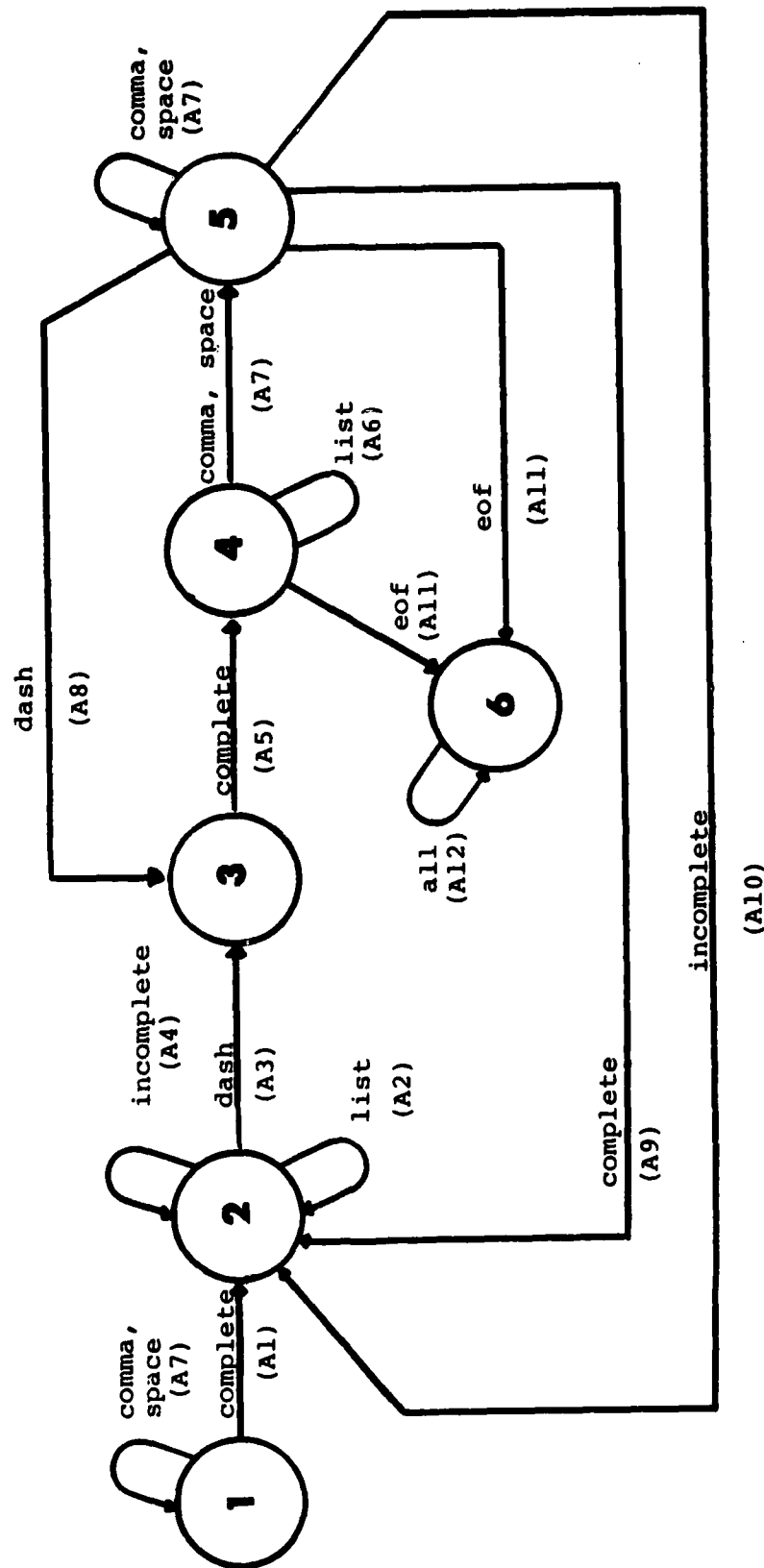
	1	2	3	4	5	6	7	8	9	10
1	2	0	0	1	0	0	0	0	0	0
2	0	2	2	0	3	0	0			
3	4	0	0	0	0	0	0			
4	0	0	4	5	0	6	0			
5	2	2	0	5	3	6	0			
6	6	6	6	6	6	6	6			
7								0		
8									0	
9										0
10										0

TRANS(*,*,2) - ACTIONS

	1	2	3	4	5	6	7	8	9	10
1	1	0	0	7	0	0	0	0	0	0
2	0	4	2	0	3	0	0			
3	5	0	0	0	0	0	0			
4	0	0	6	7	0	11	0			
5	9	10	0	7	8	11	0			
6	12	12	12	12	12	12	12			
7								0		
8									0	
9										0
10										0

NOTE: Vector elements which have been left blank are zero.

APPENDIX B



Appendix C

MACHINE DEPENDENT ROUTINES

The following routines may require modification for other machines than the TI ASC. In addition, any routines performing I/O may be susceptible to change.

NXCHAR

NUMER

ALPHA

PACK2

VALUE

BUILD

APPENDIX D
FILE INFORMATION

- FILE (01) Contains a dictionary of information on distances and angles to generate the model. (E.g. an entry of C1:S(0.77,109.5) would create an entry for an atom C with bond type S having a covalent radius of 0.77A and a covalent angle of 109.5). This dictionary can be added to or changed any time the program is run.
- FILE (02) Contains an updated version of the dictionary in file 01 and is what is output to save as the new file 01 for a future run.
- If file manipulation is a problem, these files can be eliminated and the dictionary information for each model can be supplied as part of the input data.
- File (05) Control card input.
- File (06) Printout.
- FILE (16) User supplied distance constraint information.
- FILE (26) Output file for distance constraint parameters.
 (See Figure 7.)

INPUT INFORMATION

FILE (01) File 01 is a binary file which has been written
(to file 02) and saved from an earlier MODEL RUN.

FILE (16) FORMAT(215,F7.3,3X,I1,4X,3F10.4,3X,I2)

ANO1 - First atom number

ANO2 - Second atom number

DIST - Distance between ANO1 and ANO2

TYPE - Integer code for distance type

1 = Bonded (non-hydrogen)

2 = Non-bonded (non-hydrogen)

3 = Bonded (involving a hydrogen)

4 = Non-bonded (involving a hydrogen)

5 = Intermolecular distance

XSMM1

XSMM2 - lattice translations for target atom

XSMM3

LSMM1 - Symmetry operator number for target atom

NOTE: For user added distances it will be necessary
to set the location of the "target" atom by defining
a set of lattice translations (XSMM1, XSMM2, XSMM3).

For a lattice translation of one unit cell along a,
XSMM1 will be 6.0, etc. A symmetry operator number
(LSMM1) must also be defined. This is the numerical
position of the symmetry operation in the input stream
for RESLSQ. These operations will be carried on the
coordinates of the target atom as it appears in the
input stream.

FILE (05) CONNECTIVITY DESCRIPTOR CARDS (FREE FORMAT)

** See additional description on following pages.

OUTPUT INFORMATION

FILE (02) Dictionary of bonding radii and angles - updated
and output after processing completed.

FILE (06) Printout.

FILE (26) Distance constraint information.

FILE(05) CONNECTIVITY DESCRIPTOR CARDS (FREE FORMAT)

e.g. C1:S(0.77,109.5)-C2:S,-H3:S(0.23,0.0),-H4:S

C = atom name - 2 characters are allowed and any name the user wishes is allowable as long as one is consistent. Hydrogen atoms, however, must be referred to as H because the program keys on that character to flag the different bond types.

l = position atom will occupy in CONLSQ input deck. Any numbering scheme is allowable as long as it is consistent with atom input deck to CONLSQ. A number may be used only once.

S = 1 character code for type of bond associated with a particular atom name (26 different bonding environments are possible for each atom name).

(0.77,109.5) - the bonding radius and bonding angle for the particular atom and bond type immediately preceding this information. Need be put in only once for each different atom and type of bond. A dictionary of these values can be built and used in later runs.

DELINEATORS

COLON (:)	Separates an atom name and number from a bond type indicator.
PARENTHESES ()	Enclose dictionary information. This information should be supplied from the 05 input at the first occurrence of the atom-bond type it describes (or from unit 01).
HYPHEN (-)	Separates first and second atoms describing a bond.
COMMA (,)	Indicates information about another bond follows
BLANK	A blank card ends the input deck.

NOTE: It is only necessary to describe (in the manner illustrated above) each bonded distance in the molecule. MODEL will determine non-bonded relationships and set thermal parameter constraints (according to Figure 9).

SAMPLE INPUT DECK

C1:S(0.77,109.5)-C2:S,-O9:S(0.66,109.5),-O12:S,-H16:S(0.23,0.0)

C2:S-C3:S,-N15:S(0.70,109.5),-H17:S

C3:S-C4:S,-O10:S,-H18:S

C4:S-C5:S,-O11:S,-H19:S

C5:S-C6:S,-O12:S,-H20:S

C6:S-O13:S,-H21:S,-H22:S

C7:I(0.695,120.0)-N15:S,-C8:S

C7:D(0.665,120.0)-O14:D(0.55,120.0)

C8:S-H23:S,-H24:S,-H25:S

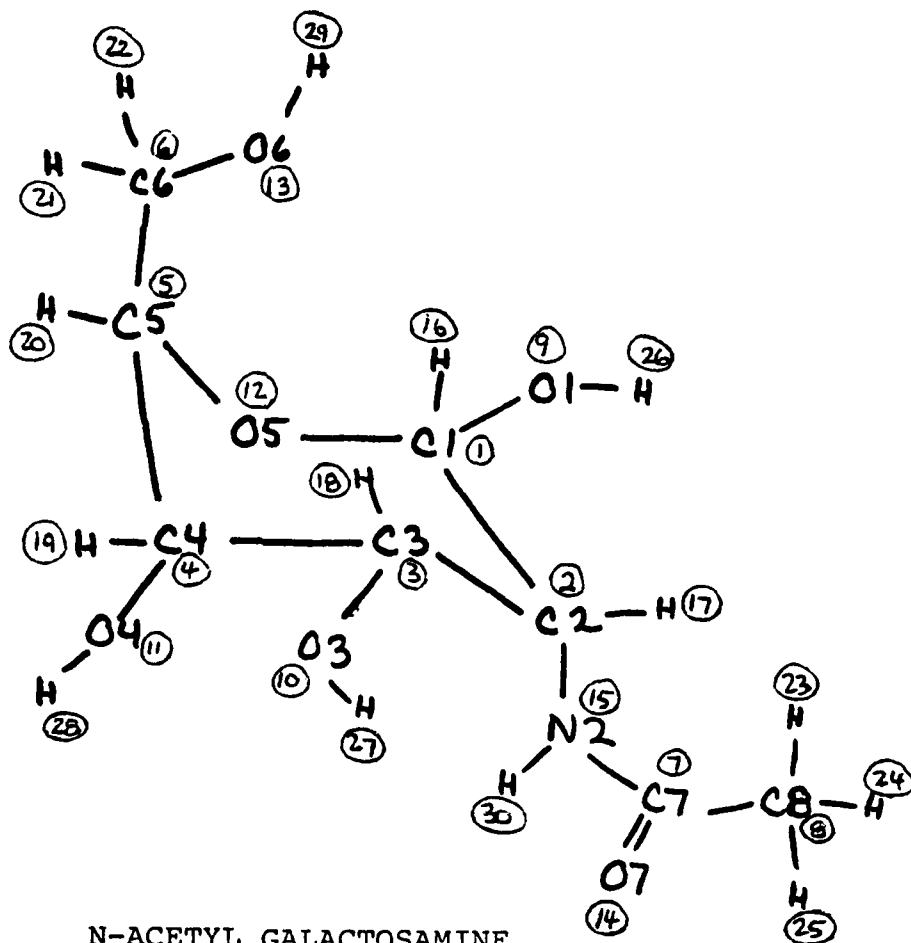
O9:S-H26:S

O10:S-H27:S

O11:S-H28:S

O13:S-H29:S

N15:S-H30:S



N-ACETYL GALACTOSAMINE

(R. D. Gilardi & J. L. Flippen (1974),
Acta Cryst., B30, 2931)

GLOSSARY

COMMON ALIST

- ATOML - (Atom list) 2-character element names for atoms in use. Set in BUILD.
- BONDL - (Bond list) subscript of first bond entry. Set in BUILD.
- AANGLE - (Atom Angle) angle of bonds for a given atom. Set in BONDD.

COMMON BONDS

- ATOMNO - (Atom number) atom numbers of target atom in bond entry. Set in BUILD.
- BNDTY1 - (Bond type 1) Bond type for originating atom. Set in BUILD.
- BNDTY2 - (Bond type 2) Bond type for target atom. Set in BUILD.
- BDIST - (Bond distance) bond distance. Set in BONDD.
- BONDNX - (Bond next) subscript of next bond entry. Set in BUILD.
- SYMM1 }
SYMM2 } → - Sets position of identity molecule relative to
SYMM3 } translations along cell edges. All three are set to 5.0 which is equivalent to a lattice translation of 0.0 along all cell edges for the origin molecule.
- LSYMM - = 1 (symmetry operator # for origin molecule)

COMMON INPUT - Used primarily in routines PARSE and BUILD.

- ANO - (atom number) Atom number of originating atom
- ANM - (atom name) 2-character element symbol.
- BT1 - (Bond type 1) 1-character bond type for ANO.

BNO - (Bonding atom number) atom number of target atom.
 BNM - (Bonding atom name) 2-character element symbol
 of target atom.
 BT2 - (Bond type 2) 1-character bond type for BNO.
 NBNM - (new bond name) 2-character symbol for new bond
 types.
 NBT - (new bond type) 1-character bond type for new bond
 types.
 NBDIST - (new bond distance) bond radius for new bond types.
 NBANG - (new bond angle) angle for new bond types.
 CHAR - (character) current input character.
 LPTR - (line pointer) current position (column) in input
 line.
 LINE - (current input line) current input line.

COMMON RADII

ELEM - (Element name) 2-character element symbol.
 RADL - (Radius list) subscript of first bond type entry
 for each entry in ELEM.
 BTYPE - (bond type) 1-character bond type
 BONDR - (bonding radius) bonding radius
 ANGLE - (bonding angle) bonding angle
 BRLNXT - (bond radius list next) link to next bond radius
 list entry.

COMMON SPS

BLSP - (bond list stack pointer) Allocation pointer for
 bond list entries.

ASP - (Atom list stack pointer) Allocation pointer
for element list.

RADSP - (Radius list stack pointer) Allocation pointer
for bonding type list.

COMMON STATE - Used in PARSE

CURST - (current state)

TRANS - (transition matrix)

COMMON COUNTS - Used in BONDD, NBONDD, and PREFIX

BDCNT - Number of bonded distances (No H)

NBDCNT - Number of non-bonded distances (No H)

XBONDS - Number of bonded distances (with H)

XNBND - Number of non-bonded distances (with H)

XINTER - Number of intermolecular distances

PROGRAM RESLSQ

A restrained least-squares refinement program using
conjugant gradient sparse-matrix methods.

Written by

John H. Konnert

Richard Gilardi

Judith L. Flippen-Anderson

Laboratory for the Structure of Matter

Naval Research Laboratory

Washington, D. C. 20375

RESLSQ

INTRODUCTION

A least-squares refinement which imposes restraints on atomic positions and on thermal parameters has proven to be extremely useful for macromolecules (Konnert, Acta Cryst (1976) A32, 614; Hendrickson and Konnert, "Aspects of Biomolecules" (1978) Ed. by R. Srinivasan, Vol. 1, Pergamon Press; Konnert and Hendrickson, Acta Cryst (1980) A36). The restraints effectively reduce the number of parameters to be determined which means, for macromolecules, that meaningful refinements can be carried out with very limited data sets. Intermediate sized systems (up to several hundred atoms), with high resolution data sets, generally refine smoothly to a definite endpoint with full-matrix least-squares techniques. However, full matrix refinement with anisotropic thermal motion is a large and expensive procedure for mid-sized molecules. The development of RESLSQ, an anisotropic sparse-matrix refinement program with distance, angle, and thermal variance restraints, provides a rapid, economical alternative.

Rapid convergence is due, for the most part, to the fact that only elements of the derivative matrix related to the physical restraints need be calculated and stored (on the order of 1% of the number of terms needed for a full-matrix refinement for a 100 atom problem). Storage requirements increase linearly, not quadratically, as the number of atoms being refined increases. In RESLSQ, all parameters are refined simultaneously,

but only correlations between bonded atoms and between atoms bonded to a third atom are normally included. Since the off-diagonal elements which represent these "bonding correlations" are dispersed (sparsely) throughout the matrix, the problem is not separable into a number of smaller sets of equations, as it is in other sparse-matrix techniques such as diagonal or block-diagonal least-squares refinement. Matrix inversion is not a space-efficient technique for sparse matrices; the inverse of a sparse matrix is not sparse. However, the conjugate gradient technique is an iterative technique which retains the sparse derivative matrix throughout the solution of the normal equations. In the programming, only the non-zero elements of the (hypothetical) $N \times N$ coefficient matrix are packed into a one-dimensional program array. A simple "bookkeeping" scheme enables the location and retrieval of any element as needed.

Hydrogen atoms may be included with anisotropic thermal factors from the onset of the refinement, since the thermal factor restraints ensure that the thermal parameters of the model will stay within reasonable stereochemical limits.

In our laboratory with mid-sized (approx. 100 atoms) anisotropic molecules, we have achieved an increase in refinement cycle speed of up to 30 times that of full matrix techniques. (Although RESLSQ may take more cycles to converge than full-matrix methods) We have also been able to do simultaneous anisotropic

refinement on all atoms in cases where the full-matrix would be too large to store on even a very large computer.

This refinement package consists of two basic programs. The first, "MODEL", constructs an "ideal" molecule from a connectivity matrix, covalent radii, and bond angles supplied by the user and then writes files of "ideal" interatomic distances towards which the "real" model will be constrained during refinement.

The second program, "RESLSQ", performs the least-squares refinement. It imposes elastic bounds on all of the key geometry-defining interatomic distances which were identified by MODEL (or specified by the user) and also restrains the thermal factor parameters in such a manner that the root mean squares amplitudes of relative motion for these atom pairs do not exceed physically reasonable values; i.e. 0.05A for covalently bonded atoms and 0.1A for next nearest neighbor atoms. How tightly these restraints are enforced is at the user's discretion and may be changed during the course of a refinement.

PHYSICAL RESTRAINTS IMPOSED ON THE ATOMIC MODEL:

1. Covalent bonds are distributed about "ideal" values.
2. Bond angles remain near "ideal" values.
3. Approximately rigid (planar, etc.) groups remain so.

Both positional and thermal parameters must be consistent with the above conditions.

THERMAL RESTRAINTS IMPOSED ON THE ATOMIC MODEL:

1. The variance of interatomic distance distributions that are related to bond lengths and angles are restrained to suitable low values. These variances are expressed in terms of individual atom thermal parameters.

NOTE: Throughout the program we are dimensioned for 500 atoms - except for AM, AMM, A and S (the latter are in SCALE and SCALEB) which are currently dimensioned (40000) large enough for ~300 atoms and ~900 distances. For more atoms or distances these arrays must be increased. They require $(21 \times NA) + (36 \times NDIS) + 1$

RESLSQ CONTROL CARDS

Type 1: (one card) FORMAT 18A4
 TITLE CARD - Any alphanumeric information

Type 2: (one card) FORMAT (8F9.4)
 Real cell dimensions

1-9 A
 10-18 B
 19-27 C
 28-36 ALPHA
 37-45 BETA
 46-54 GAMMA

Type 3: (one card) FORMAT (2I5,5A4)
 1-5 NSYM = number of symmetry operations
 6-10 NCENTR = 0 for centro space groups
 1 for noncentric space groups
 SPCGRP = 20 character field for comment on space
 group (used only for printout)

Type 4: (three cards per symmetry operation) FORMAT (4F5.0)
 Card one x
 Card two y
 Card three z
 Each card contains an x,y,z, and translation (translations are in 12th's).

e.g. \bar{x} 1/2 + y \bar{z}

Card one	-1	0	0	0
Card two	0	1	0	6
Card three	0	0	-1	0

NOTE: If NCENTR=0, omit
 symmetry operations
 related by inversion.

Type 5 (one card) FORMAT (8I5)

1-5	NA	= number of atoms
6-10	NTYP	= number of types of atoms
11-15	NDIS	= total number of distances to be restrained
16-20	NPLN	= number of planes
21-25	NV	= number of variables (NA*3 + number of scale factors)
26-30	NOBS	= maximum number of reflections to be used in refinement
31-35	NSCAT	= scattering factor table number for H atoms
36-40	IANOM	= flag to read anomalous dispersion corrections
		0 - skip
		1 - read

Type 5A: (two cards) FORMAT (8F5.3)

Read these cards only if IANOM = 0

First card - f' for each atomic scattering type

Second card - f" for each atomic scattering type

Type 6: (one card) FORMAT (I0I5)

1-5	NC	= number of cycles to be done with conditions currently being described
6-10	NOUT	= an integer value such that reflections having (Yo-Yc) greater than NOUT times SIGMAF will be considered outliers and will not be included in the refinement
11-15	NSK	= 0, add products of structure factor derivatives to the distance blocks 1, do not add...

16-20	IPRF	=	0, do not print reflections 1, print reflections 2, print only outliers
21-25	IORFE	=	punch options for conversion of thermal factors to form suitable for ORFFE or for journal publication 0, no conversion 1, punch a deck of S.D.'s suitable for input to ORFFE -1, product a listing of coordinates and big Bij's as well as ORFFE deck (coordinates as used through- out are compatible with ORTEP)
26-30	IWGT	=	0 for equal weights (i.e. all sigmas on file 20 set equal to SIGMAF) 1 for weighted refinement (i.e. all sigmas on file 20 multiplied by SIGMAF)
31-35	IDIFF	=	0, skip writing difference file 1, write file for difference map on file 30 excluding outliers 2, write file for difference map on file 30 including outliers
36-40	NRFAC	=	0, refine as usual 1, calculate R-factor only - no refinement
41-45	IDUMP	=	0, shortened printout 1, gives dump-type printout of first few matrix elements (for both distances and thermal para- meters) after restraints have been added
46-50	IHRID	=	0, refine as usual 1, set H atoms B's to be the same as those of the atoms to which they are bonded

Type 7: (one card) FORMAT (4F10.5)

1-10	FMIN	=	low FOBS cutoff (i.e. all data with FOBS<FMIN will be omitted from refinement)
11-20	SMIN	=	low sin theta/lambda cutoff
21-30	SMAX	=	high sin theta/lambda cutoff
31-40	BRAT*	=	ratio of thermal factor increase f. ~ hydrogen (over the atom to which it is bonded)
41-50	DIFCUT	=	SINTH/LAMBDA cutoff for difference map

Type 8: (one card) FORMAT (10F8.3)

1-8	SIGMAF	=	extra scaling factor for standard deviation in FOBS; if IWGT=0 then all FOBS sigmas will equal SIGMAF; if IWGT=1 than all FOBS sigmas on file 20 will be multiplied by SIGMAF
9-16	WDSKAL	=	overall multiplier for distance restraint contributions (used to adjust relative restraint weight in refinement)
	SIDG(N)	=	"ideal" standard deviations for various distance classes
17-24			N=1 for bonded distances without H
25-32			N=2 for non-bonded distances without H
33-40			N=3 for bonded distances with H
41-48			N=4 for non-bonded distances with H
49-56			N=5 for intermolecular distances

*BRAT should = 0.0 for hydrogen thermal parameter to be equal to that of its bonded atom.

goes into matrix setup which effectively keeps the bond distances elastically bound to their current values rather than to a target distance. (Tends to refine by changing torsions, keeping distances and angles somewhat fixed)

NOTE: If DAMPSC is negative then it will be applied as an overall damping factor and the damping parameters on the following two cards will be ignored.

Card two: DAMPC(N) = damping factors for coordinate shifts
N is the scattering factor reference number (i.e. a different damping is allowed for each atom type)

Card two: DAMPT(N) = damping factor for thermal parameter shifts (N has same meaning as above)

These cards must be included even if blank

NOTE: It will probably be necessary to apply some damping throughout the course of the refinement (especially for thermal parameters) due to the high correlation between scale factor shifts and thermal factor shifts.

Type 12: Continuation or termination card

Read another type 6 card here.

If NC=0 (or blank) - program stops

If NC \neq 0 then a new set of refinement conditions will be accepted and this card must be followed by type 7, type 8, type 9, and type 11 cards and a new type 12 card (it is not necessary to repeat the plane information)

LIST OF SUBROUTINES

1. BCNVRT
2. BREF
3. CALC
4. CART
5. CGSOLV
6. CGSOLI
7. COMBB
8. RESLSQ (MAIN)
9. DIAG
10. DISRF
11. ELAPSD
12. FLAG
13. FREEZE
14. HRIDE
15. INIT
16. MATMLL
17. MATMUL
18. PLANE
19. PLNRF
20. RECYCL (ENTRY IN INIT)
21. REDIST (ENTRY IN DISRF)
22. RFAC
23. RSCAL (ENTRY IN SCALE)
24. SCALE
25. SCALE B
26. SF
27. SYMMIN
28. TEST

COMMON BLOCKS

ATOMS	FILES	RECIP	TYMNG
CONST	HRIDE		VAR
CONTROL	LSQ1	REFLECT	VECTOR
DERIVB	LSQMTX	RFAC	WEIGHT
DERIVS	OTHER	STDDEV	
DIFCUT	PLANES	SYM	
DIST	REAL	TABLES	

1. BCNVRT is used to calculate thermal factors which are referred to conventional crystal axes, as opposed to RESLSQ's β 's. In RESLSQ the refinement is carried out with thermal factors (β 's) which fit the form.

$$\exp [-\beta_{11}h^2 + \beta_{22}k^2 + \beta_{33}l^2 + 2\beta_{12}hk + 2\beta_{13}hl + 2\beta_{23}kl)]$$

which are the same as ORFLS β 's. If IORFE=-1, thermal factors are converted to the form

$$\exp [-1/4(B_{11}h^2a^{*2} + B_{22}k^2b^{*2} + B_{33}l^2c^{*2} + 2B_{12}hka^*b^* + 2B_{13}hla^*c^* + 2B_{23}klb^*c^*)]$$

which we sometimes call "big B_{ij} 's". These are the coefficients currently requested by Acta Cryst. They are on the same scale as the usual isotropic B 's. BCNVRT also produces a listing of coordinates and BEQ values suitable for submission with a publication. If IORFE=1, a file of standard deviations suitable for use with ORFFE is produced. ORFFE is the Oak Ridge program for calculating distances, angles, torsions, etc.

2. BREF calculates the variance in each of the restrained interatomic distances due to the difference in thermal motion indicated by the thermal ellipsoids of the two atoms. (Details of this calculation are published: Konnert, J. and Hendrickson, W. A., Acta Cryst. A36 (1980) 344-350.

Each of these variances is weighted in accordance with the expected variance of its "distance class", and added to the overall residual being minimized in the refinement.

The variance restraint is a physically reasonable restriction, based upon experience with measurements of bond variances (e.g., electron diffraction). The variance restraint could be used with any thermal expression. BREF prints out of a summary of actual and target variances, with large discrepancies flagged.

3. CALC is basically a structure factor calculation routine. It is entered once for each reflection. It is called by SF. CALC uses table look-ups to get exponentials and sines and cosines. This will probably be quicker on most computers than using the actual functions each time.

4. CART(XX,X,NTC) takes NTC 3-component vectors, X, expressed in unit cell fractional components, and transforms them to NTC vectors, XX, which have Cartesian Angstrom coordinate components.

5. CGSOLV solves the set of linear equations, known as the normal equations, which are obtained by expanding the overall residual in a Taylor series (linearizing), and minimizing with respect to shifts in the parameters. In matrix shorthand, this set is written

$$A \vec{x} = \vec{r}$$

In the program, these letters are used to name the corresponding arrays: X(3500) is the shift vector, R(3500) is the righthand-side vector, and A(30,000) is the coefficient matrix - a sparse 2-dimensional matrix whose non-zero elements are stored in a one-dimensional computer array. The normal equations are solved by the conjugate gradient procedure (Konnert, J., Acta Cryst. (1976) A32, 614-617; Hestenes, M. R. and Stiefel, E., Journ. Natl. Bur. Stand. (1952) 49, 409-436. The notation used in CGSOLV exactly parallels that used in the cited references.

The conjugate gradient iterative procedure has been found to converge more rapidly if the normal equations are scaled in such a way that the diagonal elements are equal. This is automatically done before starting the c.g. iterations by calling subroutine scale. The unscaled shift is retrieved after the c.g. procedure by calling RSCAL (an alternate entry in SCALE).

CGSOLV is "special" rather than general mainly because of the bookkeeping scheme necessary when storing just a few 2-dimensional components in a 1-dimensional array. It will have to be modified whenever this scheme is modified (e.g., if more or less thermal parameters are used, or more scale factors, etc.).

6. CGSOL1 is almost exactly the same as CGSOLV, and one or the other could probably be dispensed with by recoding. The thermal parameter refinement is completely separable from the position refinement and certain subroutines (CGSOL1, MATMLL) deal only with the thermal refinement.

7. COMBB applies the shifts from a refinement cycle to the input parameter file (unit 10), with damping factors, if so specified, and writes the new parameter file onto unit 11. In addition, it calculates and prints the average position shift (in Angstroms) before and after damping, keeps track of the largest atomic shift and prints it out, and currently flags all atoms which move more than 0.25 Å with quadruple astericks. Also, if the scale factor changes by more than 20% of itself, the thermal factors are not refined at all that cycle. (There is a strong correlation between scale factor and the overall magnitude of the thermal factors, which can lead to lengthy oscillations in the refinement. Damping of the scale and thermal factor shifts is probably always advisable - we usually use a damping factor of 0.5 even near convergence).

8. RESLSQ is the main program, the central driving program. It doesn't really do much except sequentially call each of the major subroutines, and print out a summary of times once each cycle. It might be useful for getting an idea of the overall program flow.

Note: Its last call is to RECYL - this is an alternate entry point of INIT.

9. DIAG - Estimates the standard deviations for the parameters being varied. The S.D.'s are based only on the diagonal matrix

elements (the contributions from the restraints are subtracted out).

10. DISRF calculates the contribution of the distance restraints to the residual and to the normal matrix coefficients. Prints out a summary of actual and target distances, with large discrepancies flagged. This should be checked carefully in the beginning to guard against inadvertent connectivity errors. In later cycles, watch the actual SIGD versus the specified SIGD for the various classes. Ideally, they should roughly match. If actual sigmas are several times larger than specified sigmas, the x-ray data may be trying to tell you your target distances are not so good, and, in fact, the targets may be keeping you from a better R-factor. REDIST is an entry into DISRF which is used to skip unnecessary calculations when only an R-factor is wanted.

11. ELAPSD(N) - A call to ELAPSD will zero all eight time counters, TIME(1) to TIME(8), if N=0. (It should be initialized in this way, and is, in RESLSQ.) Later, when ELAPSD is called, the elapsed time since the last previous call is added to TIME(N). This is accomplished on our computer by a special system function, SSSET(SCCB), which returns "elapsed central processor time in seconds", each time it's called.

12. FLAG (I,N,A,B,C) - A, ideal, is compared with B, real, and divided by C, the est. stand. dev. of diff. N returns a blank, a single asterisk, or a triple asterisk depending on discrepancy magnitude.

13. FREEZE - If the HRIDE option has been chosen, then FREEZE will set the matrix and vector elements for hydrogen atom thermal factors to zero. This essentially removes the hydrogen thermal parameters from the refinement.
14. HRIDE - This subroutine is only used if IHRID=1 which indicates that hydrogen thermal parameters are not to be refined. HRIDE searches the atoms list for hydrogens and then identifies the atoms to which they are bonded. It then sets the orientation of the thermal ellipsoids for the hydrogens to be the same as those of the heavier atoms to which they bonded and it sets the ellipsoid size to be equal to or greater than (by percentage BRAT) those of the heavier atoms.
15. INIT - We have tried to localize all reading of card input in INIT (File 26 is read in DISRF). Also, most initializing of arrays and tables is done here. Entry RECYCL (near the end) rewinds and re-initializes units and arrays in preparation for further refinement. A recap of all the refinement control parameters is printed out. The comment cards embedded in this subroutine explain many of the possible refinement options available to the user.
- 16 & 17 - MATMLL and MATMULL - These routines do a simple multiplication of a matrix, A, times vector, P, to yield a vector, F. However, it is not so simple because A is not stored

as a 2-dimensional array (because it is sparse, etc...). To do the multiplication, a "bookkeeping scheme" must be used to determine which elements of A multiply the components of P. MATMUL is used for the position matrix, MATMLL for the thermal factor matrix. One would suffice, with some re-coding.

18. PLANE uses the method of Schomaker, et al. (Acta Cryst. (1959) 12, 600) to determine the best least-squares plane through a set of points.

19. PLNRF finds the deviations of atoms, in groups specified as planar, from a best least-squares plane. It calculates and prints out the individual deviations, and their contributions to the residual, and increments the normal matrix coefficients and vector elements.

20. RECYCL (see INIT)

21. REDIST (see DISRF)

22. RFAC - A "summary" routine. Subroutine SF forms 12 sums, which are (misleadingly) labelled SIG(1) to SIG(12). By combining these 12 sums in various ways to form numerators and denominators, an incredible array of R factors are formed, here in RFAC, and printed out. Fairly straight forward.

23. RSCAL (see SCALE)

24. SCALE (see also CGSOLV) - SCALE is used to perform a transformation to the normal equations:

$$A \vec{x} = \vec{r}$$

goes to

$$(SAS) (S^{-1} \vec{x}) = (S\vec{r})$$

where S is a diagonal scaling matrix, and its components are adjusted so as to equalize the diagonal components of

$$A' = SAS$$

The conjugate gradient procedure is used to solve the new set:

$$A'\vec{x}' = \vec{r} \quad \text{where } \vec{x}' = S^{-1} \vec{x} \\ \text{and } \vec{r} = S\vec{r}$$

Finally, the desired shift, x , is retrieved by calling entry RSCAL and calculating the product:

$$\vec{x} = S\vec{x}'$$

By equalizing the diagonal components, SCALE transforms to a set of normal equations in which the derivative of the parameters employed in the Taylor expansion are re-weighted so that they have equal average values. Thus equal components in the S^{-1} vector have equal structural significance. This aids in the convergence of the conjugate gradient solvent to the normal equation, since initial iterations determine the largest components of $S^{-1}\vec{x}$ most accurately.

25. SCALEB - Same operations as SCALE but for thermal shifts.

26. SF - Subroutine SF reads the reflection data from unit IREFR. It checks to see whether a reflection is to be included in the refinement. If it is, CALC is called to compute FCALC. Weights are applied here and the various agreement sums are incremented. SF calculates phases and writes the difference map (or structure

factor listing) file as desired. SF also does the derivative calculations and increments the matrix and vector elements in their respective arrays. (This part of SF is skipped when only an R-factor is desired.)

27. SYMMIN - The main function of this subroutine is the calculation of some Cartesian-based symmetry operation rotation matrices, labelled ARA (3,3,NS), which later are used in CALC to determine the ellipsoid orientation matrices for symmetry mates of the basic atom deck.

28. TEST - Checks to see if any thermal factors have gone negative and, if so, flags them but does not stop the refinement.

RESLSQ INPUT FILES

FILE 05 CONTROL CARDS (described elsewhere)
FILE 10 ATOMIC COORDINATES (read in SUBROUTINE INIT)

- 1.) scale factor card FORMAT (F10.5)
- 2.) coordinate cards - 2 per atom

Card 1: FORMAT (A8,1X,5F9.6)

ATOM - atom name
FMULT - scattering factor reference
OC - occupancy
XYZ(1) - x
XYZ(2) - y
XYZ(3) - z

Card 2: FORMAT (6F9.6)

BETA(1)
BETA(2)
BETA(3)
BETA(4) - thermal parameters
BETA(5)
BETA(6)

FILE 20 REFLECTION AND SCATTERING FACTOR DATA
(Read in SUBROUTINE SF)

BINARY FILE containing

IH1 - h
IH2 - k
IH3 - l
YO - F OBSERVED
SIGG - WEIGHT
STHOL - $\sin \theta / \lambda$
FII(NNG) - scattering factor
(one for each type of atom in the molecule)

NOTE: Each reflection carries its own scattering factor information. This helps speed up structure factor calculations.

FILE 26

DISTANCE RESTRAINT PARAMETERS (read in SUBROUTINE
DISRF) BINARY FILE

NOR - ATOM NUMBER OF ORIGIN ATOM
NT - ATOM NUMBER OF TARGET ATOM
DIS - IDEAL DISTANCE BETWEEN NOR AND NT
NWTD - BOND TYPE
1 = BONDED DISTANCE (NON-H)
2 = NON-BONDED DISTANCE (NON-H)
3 = BONDED DISTANCE (INVOLVING H)
4 = NON-BONDED DISTANCE (INVOLVING H)
5 = INTERMOLECULAR DISTANCE

XSMM1
XSMM2 LATTICE TRANSLATIONS FOR TARGET ATOM
XSMM3

LSM1 SYMMETRY OPERATOR NUMBER FOR TARGET ATOM

RESLSQ - OUTPUT FILES

FILE 06 PRINT OUT

FILE 07 (Optional; written by BCNVRT)

Output file of standard deviations for coordinates

1 card per atom: FORMAT (A8, 19X, 3F9.6)

atom name

SDX

SDY

SDZ

FILE 11 (written by COMBB)

Output file for shifted coordinates. Scale factor

is written on the first card, then two cards per

atom - same format and variables as INPUT FILE 10

FILE 30 (written by SF) FORMAT (1X,3I4, 5F11.4,11X,F11.4,
7X,F10.5)

Structure factor file

IH(1) h

IH(2) k

IH(3) l

YO FOBS

YC FCALC

APHAS A

BPHAS B

DY FOBS-FCALC

SIGYOA WEIGHTED (FOBS-FCALC)

XF PHASE (IN RADIANS)

The ASC computer and possible machine-specific code

I. Functions used

I = .AND (L,M) Performs a "Boolean and" operation on L and M.

[(I = L. AND .M) for VAX version of the program.]

I = MINO (L,M) Finds lowest of integer arguments (L,M) and places it in I.

AMIN1 (A,B) Finds lowest of floating point arguments.

The ASC has a 32 bit word. I don't think this is highly pertinent to the operation of the code in RESLSQ, except that roundoff error is appreciable, and one shouldn't expect to reproduce test outputs exactly.

The ASC has a vectorizing capability - a special compiler and hardware for processing calculation on arrays rapidly. This has affected (sometimes more than necessary) the way we set up our calculations in FORTRAN. Simple DO-loops without IF-checks will generally vectorize (and if the loop is fairly long, this can sometimes increase speed by a factor of 20 or 30!). Thus, the coding may sometimes seem curiously strung out into many successive DO-loops, when it might be more compact. This shouldn't hamper its operation, and probably not its speed, on other compilers or computers.

ASC memory is large and cheap at present - so the program is lavishly dimensioned. Besides lowering dimensions, it might be possible to do away with small arrays through re-coding (See SCALE). Also, more overlaying of space with EQUIVALENCE statements may be possible.

Implied DO-loops - In ASC FORTRAN, one can, within certain limits, use array names without explicit subscripts in arithmetic statements and the statement will be interpreted as applying to the whole array.

Example

```
DIMENSION A(900), B(900)
```

```
B = 0.0
```

```
A = B + 2.0
```

The first statement is equivalent to

```
DO 10 J = 1,900
```

```
10 B(J) = 0.0
```

and the second

```
DO 20 J = 1,900
```

```
20 A(J) = B(J) + 2.0
```

We have tried to avoid using special FORTRAN; however, the above feature is so convenient that it might possibly occur somewhere. If so, the effects may be treacherous. We would like to know if any instances are discovered.

ARRAYS IN SUBROUTINE DIMENSIONS

New Locations

BRF:	T11(3,6)	18
	AOR(3,6)	18
	AP(3,6)	18
	DP(3,3)	9
	DIR(8)	8
	NSUM(8)	8
	RMSD(8)	8

95

BCNVRT:	B11(500)	EQUIVALENCE AM(1)
	B22(500)	EQUIVALENCE AM(501)
	B33(500)	EQUIVALENCE AM(1001)
	B12(500)	EQUIVALENCE AM(1501)
	B13(500)	EQUIVALENCE AM(2001)
	B23(500)	EQUIVALENCE AM(2501)
	B11SD(500)	EQUIVALENCE AM(3001)
	B22SD(500)	EQUIVALENCE AM(3501)
	B33SD(500)	EQUIVALENCE AM(4001)
	B12SD(500)	EQUIVALENCE AM(4501)
	B13SD(500)	EQUIVALENCE AM(5001)
	B23SD(500)	EQUIVALENCE AM(5501)
	IDX(3,500)	EQUIVALENCE AM(6001)
	BEQSD(500)	EQUIVALENCE AM(7501)

0

New Locations

CALC:	TT(3,3,500)	9 x NA	
	ARG(500)	NA	
	DFDS(18)	SC	
	COS1(500)	NA	
	SIN1(500)	NA	
	FI(500)	NA	
	INDX(500)	NA	
	DA(3,500)	3 x NA	
	TF1(500)	NA	
	TFX1(500)	NA	
	HH(30)	3	

19xNA + SC + 3

CART:	XX(3,500)	3 x NA	
	X(3,500)	3 x NA	CART(XX,X,NTC)

0

CGSOLV:	P(1518)	3 x NA+SC	EQUIV(DADX(1,1))
	F(1518)	3 x NA+SC	EQUIV(DBDX(1,1))

0

CGSOL1:	P(3001)	6 x NA+SC	EQUIV(DADX(1,1))
	F(3001)	6 x NA+SC	EQUIV(DBDX(1,1))

0

COMBB:	XYZN(3,500)	3 x NA	EQUIV(DADX(1,1))
	BETAN(3,500)	3 x NA	EQUIV(DADB(1,1))
	1FLAG(500)	NA	EQUIV(DEDX(1,1))
	BEQ(500)	NA	EQUIV(DBDB(1,1))

0

PLNRF: BK(1518) 3 x NA + SC

DEL(20) NAP PLNRF(BK)

VM(B) 3

X(3,20) 3 x NAP

DIR(3) 3

6 + 4 x NAP

RFAC: RCH(4) 4

SUM(5) 5

9

SCALE: FK(1518) 3 x NA + SC

X(1518) 3 x NA + SC SCALE(A,FK)

SCL(1518) 3 x NA + SC

*A(40000) 40000

*S(40000) 40000

40000 + 6 x NA + 6 X SC

SCALE B: FK(3001) 6 x NA + SC

X(3001) 6 x NA + SC

SCL(3001) 6 x NA + SC

*A(40000)

*S(40000)

40000 + 12 x NA + 12 X SC

NOTE: THE S ARRAY HERE IS USED SOLELY TO ENABLE VECTORIZING CERTAIN CALCULATIONS IN SCALE A AND B. IT COULD BE RE-CODED TO ELIMINATE THE 40000 LOCATIONS USED UP BY S. IN THE VAX VERSION OF THE PROGRAM THIS ARRAY HAS BEEN ELIMINATED.

*Not big enough for 500 atoms

SF:	INDX(2500)	NDIS
	DVO(2500)	NDIS
	DVZ(2500)	NDIS
	IHKL(4)	4
	NSG(10)	10

14 + 3 x NDIS

CONLSG: NO DIMENSION STATEMENTS

DIAG: NO DIMENSION STATEMENTS

DISRF:	DSUM(8)	8
	NSUM(8)	8
	RMSD(8)	8
	DIR(6)	6

30

ELAPSD: SCCB(3)

3

FLAG: NO DIMENSION STATEMENTS

INIT:	SPACGRP(5)	5
-------	------------	---

5

MTAMLL:	P(1518)	3 x NA + SC
	F(1518)	3 x NA + SC

MATMLL(P,F)

0

MATMUL:	P(1518)	3 x NA + SC
	F(1518)	3 x NA + SC

MATMUL(P,F)

0

PLANE:	G(3,3)	9
	X(3,20)	60
	XS(3)	3
	XXS(3,3)	9
	ADJ(3,3)	9
	B(3,3)	9
	A(3,3)	9
	VM(3)	3
	VMI(3)	3
	BV(3)	3

117

SYMMIN:	TEMP(3,3)	9
	AS(3,3)	9
	SI(3,3)	9

27

UNIT:	XXX(3,500)	3 x NA
	DIST(500)	NA

4 x NA

TOTAL LOCATIONS CURRENTLY USED BY DIMENSIONS STATEMENTS

41 x NA =	20500
19 x SC =	95
4 x NAP =	120
3 x NDIS =	7500
MISC	80309
	<hr/> 108524

REFERENCES

- Busing, W. R., Martin, K. O., Levy, H. A., Ellison, R. D., Hamilton, W. C., Ibers, J. A., Johnson, C. K., & Theissen, W. E. (1975) ORXFLS3. Oak Ridge National Laboratory, Oak Ridge, Tennessee.
- Flippen-Anderson, J. L. (1982). American Crystallographic Association Winter Abstracts, Gaithersburg, Maryland.
- Flippen-Anderson, J. L., Konnert, J. H., & Gilardi, R. (1982). Acta Cryst. B38, 2865.
- Flippen-Anderson, J. L., Gilardi, R., Karle, I. L., Frei, M., Opella, S., Gierasch, L. M., Goodman, M., Madison, V., & Delaney, N. G. (1983). J. Am. Chem. Soc., submitted for publication.
- Furey, W. Jr., Wang, B. C., Yoo, C. S., & Sax, M. (1979). Acta Cryst., A35, 810.
- Hendrickson, W. A., & Konnert, J. H. (1979) in Biomolecular Structure, Conformation, Function, and Evolution, edited by R. Srinivasan, Vol. 1, pp. 43-57. New York: Pergamon Press.
- Konratko, R. B., & Lipscomb, W. N. (1982). J. Mol. Biol. 160, 265.
- Karle, I. L., Karle, J., Camerman, A., & Camerman, N. (1983) Acta Cryst., submitted for publication.
- Konnert, J. (1976). Acta Cryst. A32, 614, 1976.
- Konnert, J. & Hendrickson, W. A. (1980). Acta Cryst. A36, 344.
- Reese, D. & Lipscomb, W. N. (1982). J. Mol. Biol., 160, 475.
- Sielecki, A. R., Hendrickson, W. A., Broughton, C. G., Delbaere, L. T. J., Brayer, G. D. & James, M. N. G. (1979) J. Mol. Biol. 134, 781.